

JC20 Rec'd PCT/PTO 1 3 MAY 2005

**METHOD AND APPARATUS FOR PROVIDING ON-LINE GAME****TECHNICAL FIELD**

5

The present invention relates to an on-line game apparatus, and in particular, to an apparatus and method for an on-line role playing game wherein multiple users can play games by manipulating their characters (hereinafter "user character") individually to interact each other in an on-line virtual world (hereinafter "virtual world").

10

**BACKGROUND ART**

In conventional role playing games, individual users play games by manipulating their characters individually and interacting only with non-playing characters (hereinafter "NPCs"), which are provided in virtual worlds provided by game servers, independent of other users or user characters. NPCs include, for example, "monsters", which user characters should fight against to proceed to a next stage. In some existing role playing games, a user can create multiple characters of his/her own and develop the characters' abilities for each user character to play in the virtual world. The ability includes social class that a user character belongs to (e.g. plebian, knight, lord, etc.), estate (e.g. cyber money, weapon, and other items) and the like. A user character's activity in a virtual world is determined by the user character's ability.

20

Battle-net games, such as "Starcraft" available from Blizzard, Inc., are similar to on-line games in that multiple users meet in virtual world and can play games simultaneously through on-line, but different in that they are mere one-time match games and do not have continuity with previous games. Unlike Battle-net games, on-line games have continuity from one game to the subsequent game. Thus, in on-line games, the ability of a user character in a virtual world is typically determined by the result of games the user has played up to the previous game and the ability is carried over to the next game.

25

30

However, the prior art has some disadvantages which are generally recognized in the industry. In conventional on-line games, the number of users who can play the game is limited by the capacity of the server. When the number of users exceeds the hardware capacity, a system with a plurality of servers is chosen by

35

However, a virtual world provided by a server is totally different from and independent of those provided by other servers. Virtual worlds provided by a plurality of servers form a "parallel virtual world" in which each virtual world provided by a server is different from the other virtual world provide by a different server. Therefore, in conventional on-line games, users using different servers cannot meet and play with one another in one virtual world although they are playing the same kind of game. As users increase, however, users would like to test their capability in the virtual world to know their ability position, and furthermore they want to test their capability in other virtual worlds provided by other servers. Unfortunately, conventional on-line game systems until now cannot satisfy such desire.

However, it is not so simple to connect servers to solve the problem. Since each world provided by each server is an isolated world, which is politically, economically and socially balanced, the order of the world would be destroyed if interaction between the users in different servers were allowed without any proper arrangement.

Yet another disadvantage still exists in conventional on-line games. Some conventional on-line games employ PvP (Player vs Player) system, wherein user characters match against one another. Even though PvP system has an advantage in its high reality over other games, deriving lively participation from users, serious problems have been raised. Some users plunder other user characters' items, abusing PvP system and disturbing order in virtual worlds. Moreover, some users purchase other user characters' items in the real world instead of developing his/her own character in a virtual world. Therefore, it is being suggested that a certain limitation on PvP system is necessary to some degree. Limiting PvP system, however, leads to depriving user characters' freedom, damaging the spirit of Internet game that user characters can proceed with the game by interacting with one another.

Internet on-line games have been developed over the years in an attempt to share a same virtual world, but on-line games crossing virtual worlds have not yet been feasible.

#### **DISCLOSURE OF THE INVENTION**

In view of the foregoing, it is a primary object of the present invention to provide a noble system and method for on-line games, which can provide interaction among users who belong to different virtual worlds provided by different servers.

At the same time the present invention can take the merits of PvP system while removing the ill effects thereof, to solve the above-mentioned problems.

It is also an object of the present invention to provide a game apparatus, which can provides interaction between users in different servers without affecting  
5 each user character's ability in the virtual world the user character belong to.

Further, it is an object of the present invention to provide a system for on-line games, which is capable of classifying user characters' abilities into a plurality of levels and allowing interaction among users at the same level.

Consistent with the foregoing objects, an on-line role playing game  
10 apparatus connected to a plurality of servers, each of which can provide an independent virtual world, is disclosed in one embodiment of the present invention as including (a) a means for allowing an access from a user, (b) a means for receiving information on said user's user character, from a server which stores information on the user character of the user who is allowed to access, (c) a means for repeating said  
15 (a) and said (b) for a plurality of users, and (d) a means for providing on-line games between the plurality of users' user characters.

Further consistent with the foregoing object, the present invention of an on-line role playing game apparatus provides a method for providing on-line games by coupling a plurality of servers, each of which provides an independent virtual space,  
20 the method comprising (a) allowing an access from a user, (b) receiving information on said user's user character, from a server which stores information on the user character of the user who is allowed to access, (c) repeating said (a) and said (b) for a plurality of users, and (d) providing on-line games between the plurality of users' user characters.

Still further consistent with the foregoing object, the present invention of an on-line role playing game apparatus provides a computer readable medium storing computer executable instructions for providing on-line games by coupling a plurality of servers, each of which provides an independent virtual space, the instructions  
25 executing the steps of (a) allowing an access from a user, (b) receiving information on said user's user character, from a server which stores information on the user character of the user who is allowed to access, (c) repeating said (a) and said (b) for a plurality of users, and (d) providing on-line games between the plurality of users' user characters.  
30

The foregoing and other objects and features of the present invention will  
35 become more fully apparent from the following description and appended claims, taken in conjunction with the accompanying drawings.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

Understanding that these drawings depict only typical embodiments of the invention and are, therefore, not to be considered limiting of its scope, the invention will be described with additional specificity and detail through use of the accompanying drawings in which:

Fig. 1 is an illustration that shows an on-line games system of the present invention;

Fig. 2 is a block diagram showing a configuration of a client;

Fig. 3 is a schematic diagram depicting a process through which game programs are updated by a client;

Fig. 4 is a block diagram showing a configuration of a connection server;

Fig. 5 is a flow chart showing a method for providing on-line games;

Fig. 6 is a flow chart depicting the process of selecting channel in more detail; and

Fig. 7 is a flow chart showing operation in the waiting room shown in Fig. 5.

### **BEST MODE FOR CARRYING OUT THE INVENTION**

20

It will be readily understood that the components and steps of the present invention, as generally described and illustrated in the Figures herein and accompanying text, could be arranged and designed in a wide variety of different configurations while still utilizing the inventive concept. Thus, the following more detailed description of the preferred embodiments of the system and method of the present invention, as represented in Figs. 1 through 7 and accompanying text, is not intended to limit the scope of the invention, as claimed, but it is merely representative of the presently preferred embodiments of the invention.

The present invention can provide basic on-line games by servers of which each of them provide an independent virtual world, and at the same time users in each servers can play on-line games among users through combined servers. A basic form of on-line games between users who belong to independent servers is a team match, while it is also possible for each team to fight against NPCs. To balance the fighting powers of teams, a plurality of game channels are provided according to user characters' abilities and sometimes penalties are imposed. Thus, according to the present invention, users who belong to independent servers can play

on-line games together and confirm their own positions in virtual worlds. Since the result of the game according to the present invention does not affect user characters' abilities in each server the users belong to, the users can enjoy the merits of PvP games without the risk of losing the abilities or items they have developed in each server environment so far, which solves many problems caused by PvP in each independent virtual space. Thus, the technical configuration of the present invention that provides new types of on-line games by coupling independent servers, can be applied, for example, to "Lineage" available from NC Soft Inc., and to other types of on-line games by those skilled in the art.

10 The method and apparatus for managing on-line games according to the present invention is described below in detail with reference to Figs. 1 to 7. The presently preferred embodiments of the invention will be best understood by reference to the drawings, wherein like elements or steps are designated by like numerals throughout.

15 Fig. 1 shows an overview of a configuration of a system for providing on-line games according to a preferred embodiment of the present invention. System (100) comprises a plurality of clients (1a - 1n) used for users' access to the on-line game, and an on-line game server portion (5) which plays a central role in the on-line game. The plurality of clients (1a - 1n) and the on-line game server (5) are  
20 connected through network which can support game protocol, such as Internet, LAN, wireless, telephone network, and other network connections.

Each client corresponds to, for example, a game client installed on a PC. A user accesses the on-line game server (5) through one of clients (1a - 1n) and sends orders for manipulating a user character. The clients (1a - 1n) access the game  
25 server portion (5) through network (3). The game server comprises general servers (501a - 501n) and connection servers (503a - 503n). The general servers (501a - 501n) provide on-line games, and each of the servers is capable of managing an independent virtual world. Each general server can provide multiple accesses from thousands of users and provide each user the same virtual world. Users can  
30 experience only the world provided by the server they belong to, but not the worlds provided by other servers.

The connection servers (503a - 503n) provide interaction among the general servers (501a - 501n) of which each provides an independent virtual world for users accessing the general server (501a - 501n). Through the connection servers (503a -  
35 503n), users can interact or fight with other users who are accessing the connection server (503a - 503n) through a different general server.

Configurations of clients (1a - 1n) and server portion (5) are described below in detail with reference to Fig. 2 to Fig. 4.

Now referring to Fig. 2 showing a configuration of clients (1a - 1n), the clients (1a - 1n) can be created by, for example, installing on-line game programs on PCs. As shown in Fig. 2, the clients (1a - 1n) comprise a game client (210) for processing orders from a user and server (5), a updating client (220) for updating user's game program, and a communication module (230) for communicating data with the server portion (5) through network (3).

The game client (210) performs functions that should be performed by user-side computers for game proceeding, including rearranging a user screen according to information received from a server and sending input received from a user to a server.

The updating client (220) receives updating information from the server (5) and updates the game client (210). For example, a user may add new virtual worlds, characters and items which did not exist when he purchased the game. If the updating client (220) receives information for displaying such virtual worlds and characters, the updating client (220) updates the game client (210), enabling the game client to translate new information.

Fig. 3 shows a process of updating game programs with an updating server (310) (not shown in server portion (5) in Fig. 1) and the updating client (220). Once a user starts a game, the updating client (220) starts and accesses the server (5) prior to game client (210). The updating client (220) sends a time stamp of a last file received from the server (5) to the updating server (310). The updating server (310) uses the time stamp to determine which files to send to the client and then sends them. In other words, the files that were updated after user-side programs were last updated are sent from the server to the client. The updating client (220) receives all necessary files and installs them on user computer, enabling the game client (210) to recognize new files. After the updating process ends, the game client (210) can translate orders from the server (5) about new virtual worlds and characters and take appropriate actions.

The communication module (230) communicates through, for example, Internet. All communication with servers is performed through the communication module (230). In other words, both input from users to servers and information from servers to game clients are transmitted through communication module (230). In a preferred embodiment of the present invention, the communication module may support TCP/IP(Transmission Control Protocol/Internet Protocol) specification.

Fig. 4 is a block diagram that shows configurations of the general servers (501a - 501n) and the connection servers (503a - 503n). Both general servers (501a - 501n) and connection servers (503a - 503n) comprise a game management main module (420), an NPC module (430), an NPC DB (435), a game management DB (445), an authentication module (460), and a user DB (455). In the general servers (501) and the connection servers (503), however, information stored in each DB and functions of each module is totally different. First, components of the general servers (501a - 501n) are described below, followed by description of components of the connection servers (503a - 503n).

10 A NPC module (430) is responsible for generating NPCs according to certain rules. The NPC DB (435) stores various information on NPCs, which includes, for example, general information on NPCs such as each NPC's roles, possible attack patterns, related items, shapes and the like, and information on the number, kinds and locations of activated NPCs. The information in a temporary file or in a database  
15 file can be stored in a storage device, such as Random Access Memory (RAM), hard disk, flash memory and others. A preferred embodiment of the present invention, however, is configured to store information on activated NPCs in a server memory, rather than in files or DB, to speed up game proceeding.

The game management DB (445) stores various data on game proceeding, such as information on game environments, user characters, user characters' current  
20 locations, each character's points (ability values) and items. When a game is over, each game management DB (445) in the general servers (501a - 501n) stores information on the virtual world changed by the result of the game, enabling users to proceed with games with continuity in virtual worlds.

25 The user DB (455) stores various information of users accessing the general servers (501a - 501n). When the user accesses the server, the authentication module (460) receives an identification code and a password entered by the user and verifies whether the user is legitimately registered. Since each general server (501a - 501n) provides an independent virtual world, each general server only stores in its user DB  
30 (455) private information of the users it manages.

The game management main module (420) coupled with a corresponding client (1) manages the entire game. Since each of general servers (501a - 501n) is independent, each game management module (420) only manages the games in the server it resides.

35 Still referring to Fig. 4, a configuration of the connection servers (503a - 503n) is now described. Since the connection server (503) provides a virtual world

wherein user characters accessing different servers can meet and play with one another, the connection servers (503a- 503n) are different from the general servers (501a - 501n) in view of information stored in DB, operation of the game management main module, and others. The NPC module (430) and the NPC DB (435), however, perform almost the same functions as in the general servers (501). Although NPCs provided by the connection servers (503) may be totally different from those provided by the general servers (501), the connection servers (503) can be synchronized with NPC modules and NPC DBs of the general servers (501) to use the NPCs of the general servers (501).

The communication module (410) provides communication between the clients (1a - 1n) and the general servers (501a - 501n) through network.

The game management DB (445) in the connection servers (503a - 503n) stores various information obtained during game and also results of a game. While a game proceeds, the game management DB (445) operates almost the same way as in general servers. However, since a main purpose of the connection servers (503a - 503n) is not providing virtual worlds with continuity, but providing meeting places for users from the general servers (501a - 501n) can compete with one another with their own abilities, information stored in the game management DB (445) can be deleted upon the conclusion of a game and only necessary information, such as each user's win-loss result and others, can be stored and managed continuously.

Since the user DB (455) and the authentication module (460) in the connection servers (503a - 503n) should be able to allow accesses from all users registered with the general servers (501a - 501n) and provide games among users, it is desirable to have the user DB (455) and the authentication module (460) manage information on all users. Although the connection servers (503a - 503n) may store and manage in its own user DB (455) information of all users registered with the general servers (501a - 501n), the connection servers (503a - 503n) may be synchronized with each user DB in the general servers (501a - 501n) to manage user information. Although the authentication module (460) may verify user authentication using the information stored in the user DB (455) of the connection servers (503a - 503n) when the connection server (503a - 503n) receives an identification code and a password from a user, the connection servers (503a - 503n) may send the user information to the general servers (501a - 501n) for user authentication through the authentication module (460) in the general servers (501a - 501n).

The game management module (420) in the connection servers (503a - 503n)



manages the whole games at the connection servers (503a - 503n). The game management main modules (420) in the connection servers (503a - 503n), however, are synchronized not only with the clients (1a - 1n) but also with the general servers (501a - 501n) through network, differently from those in the general servers (501a - 501n), since the game management main modules (420) in the connection servers (503a - 503n) should provide connections between general servers (501a - 501n). Once a game starts through one of connection servers (503a - 503n), the game management main module (420) in the connection servers (503a - 503n) is synchronized with the game management main modules (420) in the general servers (501a - 501n) and requests information to the general servers (501a - 503n), such as who joined the game and other viable information. In response to the request, the game management main modules (420) in the general servers (501a - 501n) retrieve their own game management DBs and send corresponding user character information to the game management main module (420) in the connection servers (503a - 503n). The game management main module (420) in the connection servers (503a - 503n) stores the user character information in its game management DB (445). It also performs the function of determining the level of game each user can join, which is described in more detail below.

Now referring to Fig. 5, a flow chart shows a game proceeding according to a preferred embodiment of the present invention.

For example, if a user runs a game program on his PC, a client accesses the connection server (503a - 503n) (S10). Once the client and the connection server (503a - 503n) are connected, the updating client (220) in the client server (1a) can access the updating server (310) and receives updated program components (S15). When the updating is over, the game client (210) is connected to the connection server (503a - 503n) which in turn requests user authentication (S20). When the user enters an identification code and a password in response to the user authentication request, the authentication module (460) retrieves information in the user DB (455) to verify whether the user is legitimately registered. When the user is authorized, the user is allowed to access the game. The user, who is now connected to the connecting server (503a - 503n), chooses the general server (501a - 501n) through which the user is accessing the connecting server (503a - 503n), and character he will use. After the selection, the game management main module (420) in the connection server (503a - 503n) is synchronized with the game management main module in a selected general server of the general servers (501a - 501n) to receive corresponding user information. The received character

information is stored in the game management DB (445) in the connection server (503a - 503n). The game management main module (420) evaluates the received user character ability and stores the information together with the received character information. The user character ability is used for adjusting game balance when the user selects a channel.

After selecting a user character, the user can select the channel of the virtual world he will enter (S30). Fig. 6 shows the process of selecting a channel in more detail. Since channels are classified according to user characters' abilities, users can play games in the virtual worlds adapted to their characters' abilities. Since the connection servers (503a - 503n) provide meeting places where users who have grown in independent virtual worlds can meet one another, abilities of the users who access the connection servers are not known to one another. They are not balanced and totally different. Users can, however, enjoy games in the virtual worlds adapted to their character's abilities, by being provided the channels adapted to their characters' abilities. If a user selects a channel with a higher level than his ability, the user is very likely to play games with user characters whose abilities are superior to his character's. On the contrary, if a user selects a channel with a lower level than his ability, the user is very likely to play games with user characters whose abilities are inferior to his character's. In the latter case, a certain penalty (for example, reducing speed of user character) can be imposed on the user.

After selecting a channel, the user now proceeds to a waiting room to select a game mode. Fig. 7 is a flow chart that shows the operation in the waiting room. Now referring to Fig. 5 and Fig. 7, in the waiting room, a user can open a room, wherein he can have a conversation with other users who are connected to the connection server (503a - 503n), or join in a room already opened. Through the communication by either chatting or exchanging messages or any other viable methods, users can decide teams and game modes to proceed to the next step, a game stage (S40), before starting the game (S45). The present invention provides various forms of game modes, which are described below in more detail.

When the game is played, the game client (210) receives user inputs and sends it to the server (5) through the communication module (230). The server (5) translates the user inputs and sends orders to all clients (1a - 1e) connected to the server so that appropriate actions can be taken in response to the user inputs. The clients (1a - 1n) update game screens in response to the orders. For example, if a user clicks another user character or an NPC monster with a mouse, the game client (210) translates the information and sends it to the server (5). Then, the game

management main module (420) in the server (5) detects that the user has clicked another user character, who is an enemy, or an NPC monster and sends orders to the game client (210) so that the user character can take attacking actions. The game client (210) translates the orders from the game management main module (420) in the server (5) to display the user character's attacking another user character, who is an enemy, or a monster on the computer screen of the client. Simultaneously, the game management main module (420) in the server (5) informs all the clients (1a - 1n) the user character of an enemy or a monster being attacked, in order for the same images to be displayed.

The NPC module (430) and the NPC DB (435) are responsible for displaying NPCs. The NPC module (430) generates NPCs according to instructions from the game management main module (420) or certain rules, controls NPCs according to the actions specified in the NPC module (430), such as kicking action, punching action, and others, and allow user characters to gain points or items when the users eliminate NPCs. When the NPC module (430) informs the game management main module (420) of information on an action of a specific NPC, the main module (420) sends the information to all game clients (210) that have user characters within the visual range of the NPC, and the game client (210) translates instructions from the server (5) to display the action of the NPC. To accomplish this, the main module (420) manages the information on where user characters and NPCs are located in the virtual world.

The process of playing games with synchronization capability between the clients (1a - 1n) and the server (5) is illustrated by the example above. However, such game process can take various forms well known to those who are skilled in the art, depending on role assignments between the clients (1a - 1n) and the server (5).

In the example mentioned above, the server (5) makes almost all decisions on game proceeding while the clients (1a - 1n) only update user screens according to instructions from the server (5). In this configuration, the clients (1a - 1n) store in a user computer various graphic data and programs used for displaying game screens and use them to display game screens according to instructions from the server (5). Therefore, only the instructions defining user characters' locations and actions are transmitted through the network connected between the clients (1a - 1n) and the server (5).

For example, if a virtual world containing woods, bridges and caves, graphic information on the environments is stored in users' computers, and the clients (1a - 1n) translate instructions from the server (5) to derive and display graphic

information on user characters' locations and actions to proceed with game. Actually, displaying characters on a screen is a result of a series of events taking place in several components. After a user instruction is entered (for example, by a click of mouse on user character), the instruction can be transmitted to the server (5) through the game client (210). The instruction is translated in the server (5), and then transmitted back to the game client (210) to display revised character on the screen. Even though several steps are followed to display a revised character on the screen, a user cannot notice any time delay or hold-up in manipulating his/her user character because the process of instruction, transmission, interpretation and back transmission take place in a very short period of time through the network (3).

For the role assignment between the clients (1a - 1n) and the server (5), the server (5) and the clients (1a - 1n) should share information on the same virtual world environment. All user characters' locations are updated in the clients (1a - 1n) and the server (5) according to inputs entered by the clients (1a-1n). Moreover, since the server has the entire information on user characters and monsters' locations and appearances, it should determine what information should be displayed on each user's user screen (i.e. user's visual range) and inform each user's game client of the information. Since the server (5) sends instructions to display screens at the clients (1a - 1n) based on all information of game inputs entered by users and all information on game management, all users connected to the server (5) can play games in the same virtual world.

When a game is over, character's points and ranks are calculated based on the result of the game (S50). This information is stored in the game management DB (445) by the game management module (420) and managed with continuity of games, so users can start a next game from where he/she ended last time. When a game is over, users can move to the waiting room (S50) and users can rearrange teams, purchase items to prepare a next game, join another game, or exit if he does not want to continue.

Although a user character can be dead or loose items he has developed laboriously in a game, result of the game in the virtual worlds connection servers (503a - 503n) does not affect user character's ability in a separate servers (501a - 501n) the user character originally belongs to.

In a preferred embodiment of the present invention, a user can select among various game modes, such as a death match, a honor match or a quest battle in the waiting room stage (S35) in Fig. 5.

(i) A death match is the game mode wherein teams fight until the opponent

team is eliminated, which is the most common game mode. In a death match, there are basically two teams. When the game starts, characters are marked to be distinguished between our forces and enemy forces. When a character is dead, the user cannot take any actions except watching the game proceeding before the game is over.

(ii) A honor match is a game mode wherein teams compete for the opponent team's symbol, such as flags or crests. In the honor match, there are two teams and the team that destroys or deprives the opponent team's crest of honor first wins the game. The symbol of honor, which is an object of offence and defense, has a certain amount of energy, and the game is over when the energy of symbol becomes zero by the opponent team's offence. If a character is down or killed during the game, he could be revived to join the game again, since winning requires only destroying or depriving the crest of honor.

(iii) A quest battle is a game mode wherein user characters do not fight with one another, but with monsters. In a quest battle, there are also two games and each team passes through the same path along which monsters are disposed or different paths with similar difficulty levels. The team that passes the path first wins the game.

Since the three game modes are so configured that a plurality of users form teams, fighting powers of the teams can be unbalanced even though the users forming the teams have entered the same channel. Therefore, in order to balance objective fighting powers of both teams, a penalty can be imposed on the superior one. In an example case of battle quest, powers of the monsters disposed along the paths each team should go through or the lengths of the paths can be established differently according to the team power. In a honor match, the energy initially assigned to the symbol can be set differently. In a death match, the speed of the user characters who belong to the superior team can be reduced by a certain amount. Fighting powers of both teams can be balanced by various methods.

The forms of the above-mentioned three game modes can be one versus one or one versus a team.

To add competitiveness of users and to enhance user enjoyment of games, users who played games can be ranked. The users are granted default points when they access the connection server (503) and they are granted plus or minus points according to the game result. The users are ranked based on their points. The ranking of users are posted on user bulletin and can be monitored by all users who played the games.

The ranking can be determined in various forms. "Same level ranking" is the ranking according to character abilities based on character information. "Same server ranking" is the ranking of the users who belong to the same server. "Same channel ranking" is the ranking of the users who entered through the same channel.

5 Some other methods of ranking can be developed by those skilled in the art.

The ranking may be updated in real time or periodically with fixed time intervals.

Although the present invention has been described through preferable embodiments, various modifications and changes are also possible within the scope of the appended claims, without departing from its spirit. Specifically, the present invention can include various embodiments as follow.

10

(1) Although in the preferable embodiment of the present invention, the game played through the connection servers (503a - 503n) do not provide continuity, game continuity can be provided in another embodiment by storing information on user character when the game is over. In this case, users can select not only the characters stored in the general servers (501a - 501n), but also the characters stored in the connection server (503a - 503n). To use the characters from both servers, updated user character information, such as the items that user characters have acquired in the games provided by the connection servers (503a - 503n) can be stored in the game management DB (445) in the connection servers (503a - 503n).

15

20

(2) Although in a preferable embodiment of the present invention, the game management module (420) in servers mostly controls the game, it is also possible to move most of functions to clients in another embodiment. In this case, each connection server is only responsible for connecting users in the network while various kinds of information for game are generated by each client and broadcasted to other clients joining the game. For example, when a user character "A" attacks another user character "B," instructions for displaying this action can be generated by the user client "A," not by the server (5) and broadcasted to other clients joining the game. Each client receiving the instructions translates the instructions to perform functions to properly display characters on screen. In such modified embodiments, the connecting servers (503a - 503n) is only responsible for connecting users when the game starts and recording the game result when the game is over.

25

30

(3) The connection servers (503a - 503n) and the general servers (501a - 501n) can be implemented in a single server. Moreover, for client programs, programs for accessing the general servers (501a - 501n) and the connection servers (503a - 503n) can be implemented in a single program or in separate programs.

35

(4) Although in a preferred embodiment illustrated in Fig. 5, a user selects the server his/her character is stored, the user character can be extracted from other servers which contains user characters based on the user information obtained from the connection server (503a - 503n), without using user's own character. If a user owns a plurality of characters in a plurality of servers, user's all characters can be called in to the connection server (503a - 503n) and the user may select one them.

#### **INDUSTRIAL APPLICABILITY**

According to the present invention, it is possible for the user characters that exist in independent virtual worlds to interact and proceed with game through the virtual worlds provided by the connection servers. Therefore, users can enjoy games with other users in various environments. Moreover, since the result of game played through the connection servers does not affect user characters' abilities in the individual servers, users can play games without risking their own characters they have developed. Thus, users can enjoy matches between characters freely and the ill effects of PvP games occurring in separate servers can be mitigated by providing places for the matching.